

La enseñanza de la Gestión Proyectos de Software en el plan de estudios de Ingeniería Informática de la Universidad de Zaragoza

Rubén Bejar¹, Oscar Cantan², Pedro R. Muro-Medrano³, F. Javier Zarazaga⁴

Dept. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza

C/ María de Luna 3, Zaragoza - 50015

e-mail: ¹[rbejar](mailto:rbejar@posta.unizar.es), ²ocantan@ebro.cps.unizar.es

³[prmuro](mailto:prmuro@posta.unizar.es), ⁴javy@posta.unizar.es

Resumen

En 1997 se presentaba en las JENUI el nuevo enfoque que se la había dado a la asignatura de Proyectos en el plan de estudios de Ingeniería Informática de la Universidad de Zaragoza (ver [1]). Cuatro años después, este enfoque mantiene sus puntos básicos aunque ha sufrido la lógica evolución fruto de la experiencia tanto docente como profesional de los profesores participantes en la misma. Esta evolución también ha venido influenciada por las opiniones que sobre la asignatura, su enfoque y su utilidad han venido aportando los alumnos que la han cursado.

En este trabajo se presenta la configuración actual de esta asignatura, aspectos relativos a la evolución seguida, y las conclusiones de la experiencia de utilizar estas técnicas docentes durante estos últimos cinco años.

1. Contexto de la asignatura

En las anteriores asignaturas del bloque temático de Ingeniería del Software el alumno ha recibido formación sobre una panorámica de las distintas fases del ciclo de vida del software y acerca de los problemas, métodos y herramientas que pueden ser utilizadas en cada fase (como análisis de requisitos, diseño, producción, pruebas y mantenimiento). El planteamiento de esta asignatura es completar la formación del alumno en los aspectos técnicos y profesionales de la ingeniería del software y adquirir experiencia práctica en el desarrollo en equipo de proyectos de software. Dentro de este contexto general nos

planteamos de forma más particular el trabajo en los siguientes aspectos:

- Estudio de problemas, métodos y herramientas relacionado con las actividades de gestión que se prolongan a lo largo de todo el ciclo de vida del software: gestión del proyecto, gestión de configuraciones y aseguramiento de calidad.
- Conocer el papel y uso de los estándares de software.
- Adquirir experiencia práctica en el desarrollo completo de proyectos de software donde se cubran todas las actividades de desarrollo y gestión y se contemple un producto de software, no sólo en su componente de código, sino conjuntamente con toda la documentación de producto asociada.
- Sensibilizar al alumno en la necesidad (que surge en muchas ocasiones) del trabajo en equipo y toda la problemática que esto lleva consigo. Para ello el alumno deberá enfrentarse a los problemas de organización y comunicación de información técnica y organizativa entre los miembros del equipo. Aspectos como definición de papeles de trabajo, responsabilidad en la realización de sus tareas y comprender el problema de dirección de un equipo en toda su extensión, son también elementos importantes que tratarán de cubrirse en esta asignatura desde una perspectiva fundamentalmente práctica.
- Hacer que el alumno tenga que elegir y poner en práctica el uso de técnicas aprendidas en asignaturas anteriores (y no sólo de ingeniería del software) para resolver o

documentar distintos aspectos de un proyecto de software.

2. Métodos de enseñanza utilizados

Desde el principio hay algo de lo que estamos absolutamente convencidos y es que todas las tareas relacionadas con la construcción del software se aprenden con la experiencia. Esto es especialmente importante en una asignatura en la que los aspectos más teóricos como métodos, herramientas, formas de documentos, estándares, etc. tienen un carácter muy empírico y su explicación o análisis resulta frecuentemente aburrido. En este sentido creemos que no resulta rentable pedagógicamente que el alumno se aprenda de memoria gran cantidad de información, que por otra parte suele cambiar periódicamente (normalmente los estándares tienen una vigencia de 5 años), si no que sepan hacer uso práctico de estas informaciones apoyándose en la documentación y bibliografía necesarias.

En el planteamiento que se presentó hace cuatro años proponíamos una estrategia docente que se podía desglosar en tres partes:

- Una serie de clases teóricas, soportadas por transparencias, en las que se estudian los problemas, métodos y herramientas.
- Unas prácticas en las que se usan herramientas o calculan medidas del sistema.
- Un proyecto de asignatura en la que se desarrolla completamente un sistema de software en equipos de trabajo de 4 a 6 personas en las que uno de los alumnos tiene el papel de director del proyecto. Este proyecto es el utilizado para evaluar a los alumnos.

Aunque en la actualidad se mantiene la esencia de este planteamiento, se han introducido una serie de factores correctores destinados a buscar una mejora tanto en los resultados docentes de la asignatura, como en la aceptación de la misma por parte de los alumnos. En este sentido queremos resaltar que la introducción de estos factores no viene motivada por una sensación de fracaso en los resultados y aceptación obtenidas, sino por una búsqueda de una mejora continua de la asignatura en sí misma.

2.1. Clases teóricas

El componente teórico de la asignatura ha sido desde el principio el elemento que mayor rechazo ha causado en los alumnos. Todos los que nos movemos en el ambiente de la Ingeniería del Software sabemos que los aspectos teóricos de este mundo pueden llegar a resultar bastante pesados y difíciles de comprender. Especialmente cuando no se ha llevado a cabo la realización de proyectos software de una cierta envergadura y se ha encontrado con los problemas típicos de desarrollo y gestión de dichos proyectos.

Desde el principio hemos hecho uso de los estándares (básicamente de la ESA [2] y el IEEE) como guía para el desarrollo de los temas teóricos. Actualmente mantenemos estas guías, pero hemos tratado de descargar contenidos, centrándonos en aquellos que consideramos más básicos, y redirigiendo a una serie de referencias bibliográficas que consideramos de especial interés al alumno que desea profundizar.

Como contrapartida, hemos buscado completar estas clases teóricas con tres elementos fundamentales:

- Charlas de profesionales que trabajan en el mundo de la empresa privada y que están involucrados en proyectos software, tanto desde el punto de vista de los desarrolladores, como desde el punto de vista de los clientes. Estas charlas nos permiten validar ante los alumnos la aplicabilidad de los conceptos que les vamos presentando.
- Seminarios específicos, generalmente preparados por los propios profesores, centrados en aspectos puntuales relacionados con la gestión de proyectos software. Algunos ejemplos de estos seminarios son los que se pretende dar este año sobre la gestión de proyectos en Métrica 2 [3], y certificación de calidad en proyectos software (ISO 9000 [4] y CMM [5]).
- Ejercicios prácticos destinados a crear sensibilidad en el alumno ante problemas relacionados con la gestión de proyectos, o ante la gestión de proyectos en sí misma. Aquí nos están siendo de gran ayuda elementos extraídos del *Team Software Process* [6]. Uno de los ejemplos más representativos de los ejercicios que se plantean es el relacionado con la

organización y puesta en práctica de las reuniones de trabajo. Los resultados de este ejercicio están siendo muy satisfactorios, tanto para los alumnos, como para los profesores.

La introducción de estos tres elementos está resultando muy positiva tanto desde el punto de vista de dinamización de las clases teóricas, como desde el punto de vista de reducción del absentismo de los alumnos.

2.2. Clases prácticas

El aspecto de las clases prácticas es uno de los que más difícil de completar a nuestro gusto está resultando. Es indudable la necesidad de dotar a esta asignatura de una serie de sesiones prácticas. El problema estriba en la determinación de los contenidos de las mismas.

El planteamiento inicial fue la realización de unas prácticas centradas principalmente en el uso herramientas. Sin embargo, aquí se plantean tres problemas principales que son la determinación de que herramientas a utilizar, el coste de las licencias de dichas herramientas y la vida útil del conocimiento de dichas herramientas. El primer y segundo condicionante van muy ligados entre sí, y al presupuesto que está dispuesto a aportar el Departamento o la Universidad. Esto también influye en el tercer condicionante ya que la determinación de “la vida útil” de unas prácticas se hace en función de la disponibilidad de presupuesto para la actualización de las herramientas. Además, el trabajo de reemplazo de herramientas lleva acarreado un alto coste en revisión de las opciones existentes en el mercado y preparación del material oportuno.

Otro de los aspectos prácticos en los que se está trabajando es la realización de sesiones de seguimiento de proyectos en las que el profesor hace las veces de consultor de proceso, asesorando a los grupos en las diferentes problemáticas que se les van planteando.

2.3. Trabajo de evaluación

La asignatura se evalúa a través de un pequeño proyecto en el que se desarrolla completamente un sistema de software, desde la fase de análisis de

requisitos hasta la entrega del sistema funcionando, en equipos de trabajo de 5 personas en las que uno de los alumnos tiene el papel de director del proyecto. Al principio se dejaba libertad a que los grupos fuesen de 4 a 6 personas, pero se decidió restringir esto ante algunos problemas organizativos que se producían y que conllevaban situaciones en las que había alumnos sin grupo y era necesario ampliar alguno de estos a 7 personas.

Aunque se les da libertad a los alumnos para que compongan los grupos como quieran, queremos resaltar que esta asignatura se ubica detrás de otra que también se evalúa por trabajo y en la que los grupos deben ser obligatoriamente de 4 componentes. Esto implica la disgregación de grupos provocando la necesidad de trabajar con gente con la que no se está acostumbrado (cosa completamente habitual cuando se está en el mundo de la empresa).

El proyecto a realizar es de reducido tamaño y no tiene problemas técnicos, pero está obligado a seguir un proceso de desarrollo de razonable calidad. En los dos primeros años se estableció una única especificación de proyecto para todos los alumnos. No obstante esto acarreo bastantes problemas de comprensión del dominio de aplicación del software a construir que se solicitaba. A partir de aquí, se decidió dejar a los alumnos la posibilidad de especificar la funcionalidad del sistema a construir. El profesor juega ahora un doble papel. Por una parte determina al principio de la asignatura si el proyecto tiene el tamaño necesario (generalmente se les indica que procedan a recortar funcionalidad porque suelen pecar por exceso). Durante el proyecto el profesor hace las veces de cliente tomando como referencia las especificaciones de proyecto. Esta labor es especialmente relevante al final donde se efectúa una comprobación en la que se determina si el proyecto se ajusta a los requisitos o no. En este apartado se les pone de manifiesto las ambigüedades en las que han podido caer a la hora de especificar el sistema y que en un proyecto real podrían conducirles a quejas del cliente por incumplimiento de contrato.

En este proyecto es donde los alumnos comienzan a adquirir experiencia sobre responsabilidades, colaboración, elaboración de documentos, conocimiento del proceso y gestión. Los equipos tienen a su disposición ejemplos de los trabajos

realizados en años anteriores que les pueden servir de guía para realizar el suyo propio. Además disponen de un sistema de comunicación para preguntar cuestiones relacionadas con aspectos técnicos, de desarrollo o de gestión, directamente al profesor, o a través de las sesiones prácticas de seguimiento de los proyectos.

La revisión final del trabajo realizado se lleva a cabo mediante una auditoria del proceso y del producto. Se trata de conseguir algo más que una sesión de evaluación en la que se determina la nota que merecen los trabajos realizados. Cada grupo se reúne con un profesor por un espacio aproximado de una hora. En este tiempo se procede a revisar todo el trabajo realizado, y a poner de manifiesto las deficiencias encontradas, y el modo en el que deberían haber sido tratadas. Se trata, por consiguiente, de una última sesión teórico-práctica muy intensa que persigue que el alumno aprenda a partir de su propio trabajo.

3. Conclusión

Desde hace cinco años estamos trabajando en los aspectos relacionados con la enseñanza de la gestión de proyectos desde un planteamiento basado en el aprendizaje a través de la práctica. El esfuerzo que este planteamiento nos está implicando es muy alto. Sin embargo, consideramos que esta es la aproximación más apropiada y que permite una mayor rentabilidad para el alumno desde el punto de vista del aprendizaje. Además, cada año vemos recompensado este esfuerzo cuando los alumnos nos indican que, a pesar de todo el trabajo que para ellos les supone, la experiencia y conocimientos que han adquirido hubiesen sido difícilmente obtenidos por otros medios.

No obstante, creemos que todavía podemos mejorar los medios y los resultados. Es por ello que cada año tratamos de hacer una visión crítica de nuestro trabajo y proponemos ideas destinadas a conseguir dichas mejoras.

Referencias

- [1] P.R.Muro Medrano, J.Zarazaga Soria, J.Valiño García. "Aspectos docentes de la asignatura de Proyectos de Software del plan de estudios

de Ingeniería Informática de la Universidad de Zaragoza". Actas de las III Jornadas de Enseñanza Universitaria, (Jenui'97), páginas: 47-52, Madrid, 16-17 Junio 1997.

- [2] C. Mazza, J. Fairclough, B. Melton, D. De Pablo, A. Scheffer, R. Stevens, M. Jones, G. Alvisi. *Software Engineering Guides*. Prentice Hall, 1996.
- [3] MÉTRICA. *Versión 2.1. Metodología de planificación y desarrollo de Sistemas de Información*. Coeditado por el Ministerio de Administraciones Públicas y la Editorial Tecnos.
- [4] UNE-EN ISO 9000-3. *Gestión de la Calidad y Aseguramiento de la Calidad*.
- [5] CMU SEI, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, 1995
- [6] W.S. Humphrey, *Introduction to the Team Software Process*. Addison-Wesley, 1999.